

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Application Serial No. ....09/724,703  
Filing Date .....November 28, 2000  
Inventor..... Pradyunma K. Misra et al.  
Group Art Unit .....2193  
Examiner .....Wood, William H.  
Attorney's Docket No. ....MS1-197USC1  
Confirmation No.....9570  
Title: System and Method for Software Licensing

**APPEAL BRIEF**

To: Commissioner for Patents  
PO Box 1450  
Alexandria, Virginia 22313-1450

From: Allan Sponseller (Tel. 509-324-9256x215; Fax 509-323-8979)  
**Customer No. 22801**

Pursuant to 37 C.F.R. §41.37, Applicant hereby submits an appeal brief for application 09/724,703, filed November 28, 2000, within the requisite time from the date of filing the Notice of Appeal. Accordingly, Applicant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

<b><u>Appeal Brief Items</u></b>	<b><u>Page</u></b>
(1) Real Party in Interest	3
(2) Related Appeals and Interferences	3
(3) Status of Claims	3
(4) Status of Amendments	3
(5) Summary of Claimed Subject Matter	4
(6) Grounds of Rejection to be Reviewed on Appeal	12
(7) Argument	12
(8) Appendix of Appealed Claims	44
(9) Appendix of Evidence Submitted	60
(10) Appendix of Related Proceedings	61

**(1) Real Party in Interest**

The real party in interest is Microsoft Corporation, the assignee of all right, title and interest in and to the subject invention.

**(2) Related Appeals and Interferences**

Appellant is not aware of any other appeals, interferences, or judicial proceedings which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

**(3) Status of Claims**

Claims 1-6, 10, 13-21, 23, 24, 26, 27, 29, 45, 47-57, 59-61, 63-65, 68, 69, 71-75, and 79-88 stand rejected and are pending in this Application. Claims 1-6, 10, 13-21, 23, 24, 26, 27, 29, 45, 47-57, 59-61, 63-65, 68, 69, 71-75, and 79-88 are appealed. Some of claims 1-6, 10, 13-21, 23, 24, 26, 27, 29, 45, 47-57, 59-61, 63-65, 68, 69, 71-75, and 79-88 were previously amended. Claims 7-9, 11, 12, 22, 25, 28, 30-44, 46, 58, 62, 66, 67, 70, and 76-78 were previously canceled. Claims 1-6, 10, 13-21, 23, 24, 26, 27, 29, 45, 47-57, 59-61, 63-65, 68, 69, 71-75, and 79-88 are set forth in the Appendix of Appealed Claims on page 44.

**(4) Status of Amendments**

A Final Office Action was issued on June 16, 2005.

A Response to the Final Office Action was filed September 16, 2005. No amendments were made as part of this Response.

An Advisory Action was issued on February 22, 2006, indicating that the request for reconsideration had been considered but did not place the application in condition for allowance.

Appellant filed a Notice of Appeal on December 16, 2005 in response to the Final Office Action.

#### **(5) Summary of Claimed Subject Matter**

A concise explanation of each of the independent claims is included in this Summary section, including specific reference characters. These specific reference characters are examples of particular elements of the drawings for certain embodiments of the claimed invention, and the claims are not limited to solely the elements corresponding to these reference characters.

With respect to independent claim 1, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 32, line 15, a license pack (108) is created at a license generator (26), the license pack (108) containing a set of one or more individual software licenses. The license pack (108) is signed (160) with a digital signature of the license generator (26), and an ID of the license pack is associated with a license server (28), the ID uniquely identifying the license pack (108). This association is maintained (100) at the license generator (26). The license pack (108) is also issued to the license server (28), and the license generator's digital signature on the license pack (108) is verified at the license server (28). The software licenses contained in the license pack (108) are distributed from the license server (28) to corresponding clients (30).

With respect to independent claim 13, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 34, line 8, software licenses are distributed to clients (30) so that the clients (30) may legally execute underlying software to which the software licenses pertain. The software licenses are electronically issued as digital certificates that are distributed in one-to-one correlation with individual clients (30) and traced to an issuing authority (26), and a software license is issued to a particular client (30) only if a client executable image received from the client (30) matches a stored client executable image (120) for the particular client (30).

With respect to independent claim 15, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 32, line 15, a request (106) for a software license is received from a particular license server (28). A license pack (108) containing a set of one or more individual software licenses is created, and a license pack ID is assigned to the license pack (108), the license pack ID uniquely identifying the license pack (108). The license pack ID is associated with the particular license server (28), and the license pack (108) is digitally signed. The signed license pack (108) is issued to the particular license server (28).

With respect to independent claim 21, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 32, line 15, a request for a software license is received from a particular client (30). An authenticity of the particular client (30) is determined. This determination includes maintaining a set of client images (120), receiving a client software ID from the particular client (30), and comparing the client software ID to the client

images (120) to evaluate whether the client (30) is authentic. A software license from a pack of software licenses (108) is selected that is appropriate for the particular client (30), the software license having an associated license ID. The license ID is associated with the particular client (30), and the software license is granted to the particular client (30).

With respect to independent claim 45, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 32, line 15, a system for licensing software includes a license generator (26) and a license server (28). The license generator (26) creates a license pack (108) containing a set of one or more individual software licenses, and digitally signs (160) the license pack (108) with a digital signature. The license generator (26) also assigns a license pack ID to the license pack (108) and keeps an association (100) of the license pack ID with the license server, the license pack ID uniquely identifying the license pack (108). The license server (28) is remote from, but operatively coupled to, the license generator (26). The license server (28) receives the license pack (108) from the license generator (26), verifies the license generator's digital signature on the license pack (108), and stores the individual licenses for subsequent distribution to individual clients (30).

With respect to independent claim 55, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 32, line 15, a license generator (26) issues packs of software licenses (108) to authorized license servers (28). The license generator (26) includes a request handler (104) to receive a request (106) from a license server (28) for a license pack (108). A license producer (102) is responsive to the request (106) received by the request

handler (104) to generate a license pack (108) containing a set of one or more individual software licenses. The license producer (102) assigns a license pack ID to the license pack (108), associates the license pack ID with the license server (28), and digitally signs the license pack (108), the license pack ID uniquely identifying the license pack (108).

With respect to independent claim 59, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 32, line 15, a code segment creates a license pack containing a set of one or more individual software licenses and assigns a license pack ID to the license pack, the license pack ID uniquely identifying the license pack (152). A code segment also associates the license pack ID with the particular license server (156), and a code segment also digitally signs the license pack (160).

With respect to independent claim 60, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 34, line 8, a license server (28) issues individual software licenses from a software pack (108) received from a licensing clearinghouse (22). The license server (28) includes a license store (112), a request handler (122), a client authenticating module (124), and a granting module (126). The license store (112) stores the software pack of individual software licenses, each software license having an associated license ID. The request handler (122) receives a request for a software license from a client (30). The client authenticating module (124) determines, based on a client image received from the client (30), whether the client is authentic and can receive a software license. The granting module (126) grants a

software license from the license store (112) to an authenticated client (30), and associates the license ID with the authenticated client (30).

With respect to independent claim 68, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 34, line 8, a code segment receives a license pack (108) from a license generator (26), the license pack (108) containing a set of one or more individual software licenses. A code segment also validates the license pack (108), and a code segment also stores the software licenses (112). A code segment (124) also, responsive to a request for a software license from a client (30), determines, based on a client executable image received from the client (30), whether the client (30) is authentic and can receive a software license. A code segment (126) also grants a software license to an authenticated client (30), the software license containing a license ID. A code segment (126) also associates the license ID with the authenticated client (30).

With respect to independent claim 69, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 34, line 8, a client computer (30) includes a license cache (136), a license requestor (132), and a challenge handler (134). The a license cache (136) stores one or more software licenses, and the license requestor (132) requests a software license from a license server (28). The challenge handler (134) handles an authenticity challenge from the license server (28), the challenge handler (134) computing a challenge response that contains a client image that can be used by the license server (28) to evaluate whether the client (30) is authentic and can be licensed. Upon authentication by the license server (28) and granting of a software license,



the license requestor (132) receives the software license from the license server (28) and stores the software license in the license cache (136).

With respect to independent claim 71, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 34, line 8, a code segment (134) receive an authenticity challenge from a license server (28) that distributes software licenses. A code segment (134) also computes a challenge response that contains a client image that can be used by the license server (28) to evaluate whether the client (30) is authentic and can be licensed. A code segment (132) also stores the software license granted by the license server (28) in an event that the client (30) is deemed authentic.

With respect to independent claim 72, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 32, line 15, a license pack table (114) records information pertaining to one or more license packs (108), the license pack table (114) being indexed by license pack IDs that uniquely identify corresponding individual license packs (108), each license pack (108) containing one or more software licenses. A client assignment table (116) records information pertaining to software licenses that are assigned to clients (30), the client assignment table (116) being indexed by license IDs that identify individual software licenses, the client assignment table (116) further having the license pack IDs of the license packs from which the corresponding software licenses are issued. The license pack table (114) and the client assignment table (116) are correlated via the license pack IDs contained in each table (114, 116).

With respect to independent claim 79, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 34, line 8, a client computer (30) includes a memory (136), a license requestor (132), and a challenge handler (134). The memory (136) stores information corresponding to one or more received software licenses. The license requestor (132) requests a software license from a license server (28). The challenge handler (134) handles an authenticity challenge from the license server (28), the challenge handler (134) computes a challenge response that contains a client image that can be used by the license server (28) to evaluate whether the client (30) is authentic and can be licensed, wherein the challenge contains a random number, and the challenge handler (134) computes the challenge response by concatenating the random number with the client image to form a concatenated value and hashing the concatenated value. Upon authentication by the license server (28) and granting of a software license, the license requestor (132) receives the software license from the license server (32), and saves in the memory (136) information corresponding to the received license.

With respect to independent claim 83, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 34, line 8, a software license is requested from a license server (28). An authenticity challenge is received from the license server (28), and a challenge response is computed that contains a client image that can be used by the license server (28) to evaluate whether the client (30) is authentic and can be licensed. Upon authentication by the license server (28) and granting of a software license, the

software license is received from the license server (28) and information corresponding to the software license is stored in a memory (136).

With respect to independent claim 86, as discussed for example at p. 11, line 16 – p. 25, line 17, p. 27, line 2 – p. 28, line 16, and p. 30, line 14 – p. 34, line 8, a client computer (30) includes means for storing (drawings: 136, 212; specification: p. 9, lines 9-18, p. 23, line 22 – p. 25, line 11, and p. 30, line 14 – p. 34, line 8) information corresponding to one or more received software licenses. The client computer also includes means for requesting (drawings: 132, 220; specification: p. 9, lines 9-18, p. 23, line 22 – p. 25, line 11, and p. 30, line 14 – p. 34, line 8) a software license from a license server (28), and means for handling (drawings: 134, 196, 230, 232; specification: p. 9, lines 9-18, p. 23, line 22 – p. 25, line 11, and p. 30, line 14 – p. 34, line 8) an authenticity challenge from the license server (28), the means for handling (drawings: 134, 196, 230, 232; specification: p. 9, lines 9-18, p. 23, line 22 – p. 25, line 11, and p. 30, line 14 – p. 34, line 8) computing a challenge response that contains a client image that can be used by the license server (28) to evaluate whether the client (30) is authentic and can be licensed, wherein the challenge contains a random number, and the means for handling (drawings: 134, 196, 230, 232; specification: p. 9, lines 9-18, p. 23, line 22 – p. 25, line 11, and p. 30, line 14 – p. 34, line 8) computes the challenge response by concatenating the random number with the client image to form a concatenated value and hashing the concatenated value. Upon authentication by the license server (28) and granting of a software license, the means for requesting (drawings: 132, 220; specification: p. 9, lines 9-18, p. 23, line 22 – p. 25, line 11, and p. 30, line 14 – p. 34, line 8) receives the software license from the license

server (28) and saves in the memory (136) information corresponding to the received license.

**(6) Grounds of Rejection to be Reviewed on Appeal**

Claims 83-85 stand rejected under 35 U.S.C. §102(a) as being unpatentable over U.S. Patent No. 5,671,412 to Christiano.

Claims 1-6, 10, 13-14, 45, 47-54, 59, 68-69, and 71-75 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,745,879 to Wyman in view of U.S. Patent No. 5,671,412 to Christiano.

Claims 15-21, 23-24, 26-27, 29, 55-57, 60-61, and 63-65 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,745,879 to Wyman in view of U.S. Patent No. 5,671,412 to Christiano and in further view of U.S. Patent No. 5,138,712 to Corbin.

Claims 79-82 and 86-88 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,671,412 to Christiano in view of U.S. Patent No. 6,049,612 to Fielder et al.

**(7) Argument**

**A. Rejection under 35 U.S.C. §102(a) over U.S. Patent No. 5,671,412 to Christiano.**

Claims 83-85 stand rejected under 35 U.S.C. §102(a) as being unpatentable over U.S. Patent No. 5,671,412 to Christiano (hereinafter "Christiano").

**1. Claims 83-85**

With respect to claim 83, claim 83 recites:

A computer-readable medium, having computer readable instructions for:

- requesting a software license from a license server;
- receiving an authenticity challenge from the license server;
- computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed; and

- receiving, upon authentication by the license server and granting of a software license, the software license from the license server and storing information corresponding to the software license in a memory.

Appellant respectfully submits that Christiano does not disclose computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed as recited in claim 83.

In the June 16, 2005 Office Action at ¶ 8, p. 27, it was asserted that “A client is identified via a client image or identifier system (column 6, line 10 to column 7, line 30). In particular note column 6, line 64 to column 7, line 1.” Christiano at col. 6, line 64 to col. 7, line 1 recites “Thus, under a node-locked policy, a computer system 12 would be able to check out a license only if a license were available on the license server which matched that computer system’s node, address, serial number, user name, or other identifier”.

However, none of these items mentioned in the cited portion of Christiano is a client image. In Christiano, the “node-locked” policy allows only one computer system at a specific node on the license management system to use a designated software product (see, col. 6, lines 61-64). Thus, the computer system’s node is not a client image. A computer system’s address is just an

address, it is not a client image. A computer system's serial number is just a serial number, it is not a client image. A user name is just that – a user name; it is not a client image. The simple mentioning of "other identifier" does not teach a client image.

There is no discussion or mention in Christiano of a client image. In fact, a search through Christiano reveals that the word "image" is not present in Christiano, nor is "client image". Accordingly, Appellant respectfully submits that the identifiers listed in the cited portion of Christiano do not disclose a client image, nor is a client image disclosed elsewhere in Christiano. As such, Appellant respectfully submits that Christiano cannot disclose the client image as recited in claim 83, much less computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed as recited in claim 83.

For at least these reasons, Appellant respectfully submits that claim 83 is allowable over Christiano.

Given that claims 84 and 85 depend from claim 83, Appellant respectfully submits that claims 84 and 85 are likewise allowable over Christiano for at least the reasons discussed above with respect to claim 83.

Accordingly, Appellant respectfully requests that the §102 rejections be withdrawn.

**B. Rejection under 35 U.S.C. §103(a) over U.S. Patent No. 5,745,879 to Wyman in view of U.S. Patent No. 5,671,412 to Christiano.**

Claims 1-6, 10, 13-14, 45, 47-54, 59, 68-69, and 71-75 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,745,879 to Wyman (hereinafter “Wyman”) in view of U.S. Patent No. 5,671,412 to Christiano (hereinafter “Christiano”).

**1. Claims 1-6 and 10**

With respect to claim 1, claim 1 recites:

A computer-implemented method comprising:  
creating a license pack at a license generator, the license pack containing a set of one or more individual software licenses;  
signing the license pack with a digital signature of the license generator;  
associating an ID of the license pack with a license server, the ID uniquely identifying the license pack;  
maintaining the association at the license generator;  
issuing the license pack to the license server;  
verifying, at the license server, the license generator’s digital signature on the license pack; and  
distributing the software licenses contained in the license pack from the license server to corresponding clients.

Appellant respectfully submits that Wyman in view of Christiano does not disclose or suggest associating an ID of the license pack with a license server, the ID uniquely identifying the license pack and maintaining the association at the license generator as recited in claim 1.

In the June 16, 2005 Office Action at ¶ 8, p. 28, it was asserted that:

Third, under the rejections found in the previous Office Action, citations of Christiano and Wyman are provided illustrating the obviousness of associating a unique name of Christiano with a

server as in Wyman. Specifically owners often wish to keep track of their customers.

At ¶ 4, p. 5, it was further asserted that:

Wyman did not explicitly state associating an ID of the license pack with a license server; and maintaining the association at the license generator. Christiano demonstrated that it was known at the time of invention to provide unique identification of a license package (Figure 2b, element 24; column 9, lines 1-4; note Christiano also disclosed license packs). Wyman demonstrated that it was known at the time of invention for owner of licensed software to keep track of their customers and the software they borrow/lease (column 7, lines 14-20). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the licensing system of Wyman with license pack ID and the licensor tracking the license ID as found in Christiano and Wyman's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide an owner of software the ability to enforce use/sales policies (the motivation behind a license in general). A license ID maintained by the owner performs this function.

Appellant respectfully disagrees and submits that there is no disclosure or suggestion in Wyman in view of Christiano of maintaining, at a license generator, an association of an ID of the license pack with a license server, the ID uniquely identifying the license pack, as recited in claim 1.

As indicated above, Wyman at col. 7, lines 14-20 is cited as demonstrating that it was known at the time of invention for owner of licensed software to keep track of their customers and the software they borrow/lease. The cited portion of Wyman reads:

The license management facility cannot create a license itself, but instead must receive a license document (a product use authorization) from an issuer of licenses. As part of the overall license management system of the invention, a license document generator is provided which creates the product use authorizations



under authority of the owner of the software, as negotiated with customers.

Appellant respectfully submits that nothing in this cited portion of Wyman, even when combined with Christiano, discloses or suggests associating an ID of the license pack with a license server, the ID uniquely identifying the license pack, and maintaining the association at the license generator. The mere discussion of a license management facility receiving a license document from an issuer of licenses does not include any discussion or mention of the license document generator maintaining an association of an ID of the license pack with a license server, the ID uniquely identifying the license pack, and maintaining the association at the license generator as recited in claim 1.

Appellant respectfully submits that it would not have been obvious in view of Wyman, even when combined with Christiano, to associate an ID of the license pack with a license server, the ID uniquely identifying the license pack, and maintain the association at the license generator as recited in claim 1. In Wyman, the license server is given the authority to grant licenses on behalf of the issuer (see, col. 9, lines 61-64). Thus, the license server is entrusted by the issuer to grant licenses. The license server maintains a license data file comprising a number of license documents or licenses (see, col. 9, lines 23-26), and grants permission to requesting user nodes to use a program when authorized (see, col. 10, lines 1-19). Thus, the license management is entrusted to, and performed by, the license server in Wyman – there is no reason why the issuer would need to perform any of the license management because it has entrusted the license server to do so.

Furthermore, in the June 16, 2005 Office Action at ¶ 8, p. 27, in response to Appellant's previous assertion as cited in the June 16, 2005 Office Action that "Christiano fails to disclose package name uniquely identifying the package", it was asserted that "a name provides a unique identification, that is the purpose of a name". Christiano discusses a package license description that includes a package name field (see, col. 8, lines 57-61). The package name field stores the package name, which is the identifier of the package (see, col. 9, lines 1-2). However, the mere use of a package name does not disclose an ID uniquely identifying a license pack. A name is not necessarily unique. For example, multiple copies of the Microsoft® Word word processing application exist, each having the same files with the same file names. As such, these names do not uniquely identify the different files. There is nothing in Christiano that requires each package license description to have a unique name, nor is there any disclosure in Christiano of why each package license description would need a unique name. Thus, Appellant respectfully submits that Christiano does not disclose an ID uniquely identifying a license pack as recited in claim 1.

For at least these reasons, Appellant respectfully submits that claim 1 is allowable over Wyman in view of Christiano.

Given that claims 2-6 and 10 depend from claim 1, Appellant respectfully submits that claims 2-6 and 10 are likewise allowable over Wyman in view of Christiano for at least the reasons discussed above with respect to claim 1.

## 2. Claims 13-14

With respect to claim 13, claim 13 recites:

A computer-implemented method for distributing software licenses to clients so that the clients may legally execute underlying software to which the software licenses pertain, the computer-implemented method comprising electronically issuing the software licenses as digital certificates that are distributed in one-to-one correlation with individual clients and traced to an issuing authority, and issuing a software license to a particular client only if a client executable image received from the client matches a stored client executable image for the particular client.

Appellant respectfully submits that Wyman in view of Christiano does not disclose or suggest issuing a software license to a particular client only if a client executable image received from the client matches a stored client executable image for the particular client as recited in claim 13.

Appellant respectfully submits that there is no discussion or mention of a client executable image in Wyman or Christiano, much less any discussion or mention of issuing a software license to a particular client only if a client executable image received from the client matches a stored client executable image for the particular client as recited in claim 13. It appears from the June 16, 2005 Office Action at ¶4, p. 8, that Christiano is being relied on as teaching the client executable image of claim 13. However, as discussed above with respect to claim 83, Appellant respectfully submits that Christiano does not disclose or suggest a client image, and thus also does not disclose or suggest a client executable image as recited in claim 13.

Furthermore, claim 13 recites (emphasis added) issuing a software license to a particular client only if a **client executable image** received from the client matches a stored **client executable image** for the particular client. In the June 16,

2005 Office Action at ¶ 8, p. 28, it was asserted that “A client is identified via a client image or identifier system (column 6, line 10 to column 7, line 30). In particular note column 6, line 64 to column 7, line 1.” Christiano at col. 6, line 64 to col. 7, line 1 recites “Thus, under a node-locked policy, a computer system 12 would be able to check out a license only if a license were available on the license server which matched that computer system’s node, address, serial number, user name, or other identifier”.

However, none of these items mentioned in the cited portion of Christiano is a client **executable** image. In Christiano, the “node-locked” policy allows only one computer system at a specific node on the license management system to use a designated software product (see, col. 6, lines 61-64). Thus, the computer system’s node is not a client **executable** image. A computer system’s address is just an address, it is not a client **executable** image. A computer system’s serial number is just a serial number, it is not a client **executable** image. A user name is just that – a user name; it is not a client **executable** image. The simple mentioning of “other identifier” does not teach a client **executable** image.

With respect to Wyman, Wyman is not cited as curing, and does not cure, these deficiencies of Christiano.

For at least these reasons, Appellant respectfully submits that claim 13 is allowable over Wyman in view of Christiano.

Given that claim 14 depends from claim 13, Appellant respectfully submits that claim 14 is likewise allowable over Wyman in view of Christiano for at least the reasons discussed above with respect to claim 13.

### 3. Claims 45 and 47-54

With respect to claim 45, claim 45 recites:

A system for licensing software, comprising:

a license generator to create a license pack containing a set of one or more individual software licenses, the license generator digitally signing the license pack with a digital signature;

a license server remote from, but operatively coupled to, the license generator to receive the license pack from the license generator, the license server verifying the license generator's digital signature on the license pack and storing the individual licenses for subsequent distribution to individual clients; and

wherein the license generator is further to assign a license pack ID to the license pack and keep an association of the license pack ID with the license server, the license pack ID uniquely identifying the license pack.

Appellant respectfully submits that Wyman in view of Christiano does not disclose or suggest the license generator is further to assign a license pack ID to the license pack and keep an association of the license pack ID with the license server, the license pack ID uniquely identifying the license pack as recited in claim 45.

Appellant respectfully submits that, similar to the discussion above regarding claim 1, it would not have been obvious in view of Wyman, even when combined with Christiano, for a license generator to assign a license pack ID to the license pack and keep an association of the license pack ID with the license server, the license pack ID uniquely identifying the license pack as recited in claim 45. In Wyman, the license server is given the authority to grant licenses on behalf of the issuer (see, col. 9, lines 61-64). Thus, the license server is entrusted by the issuer to grant licenses. The license server maintains a license data file comprising a number of license documents or licenses (see, col. 9, lines 23-26),

and grants permission to requesting user nodes to use a program when authorized (see, col. 10, lines 1-19). Thus, the license management is entrusted to, and performed by, the license server in Wyman – there is no reason why the issuer would need to perform any of the license management because it has entrusted the license server to do so.

Furthermore, similar to the discussion above regarding claim 1, Christiano discusses a package license description that includes a package name field (see, col. 8, lines 57-61). The package name field stores the package name, which is the identifier of the package (see, col. 9, lines 1-2). However, the mere use of a package name does not disclose an ID uniquely identifying a license pack. A name is not necessarily unique. For example, multiple copies of the Microsoft® Word word processing application exist, each having the same files with the same file names. As such, these names do not uniquely identify the different files. There is nothing in Christiano that requires each package license description to have a unique name, nor is there any disclosure in Christiano of why each package license description would need a unique name. Thus, Appellant respectfully submits that Christiano does not disclose an ID uniquely identifying a license pack as recited in claim 45. With respect to Wyman, Wyman is not cited as curing, and does not cure, these deficiencies of Christiano.

For at least these reasons, Appellant respectfully submits that claim 45 is allowable over Wyman in view of Christiano.

Given that claims 47-54 depend from claim 45, Appellant respectfully submits that claims 47-54 are likewise allowable over Wyman in view of Christiano for at least the reasons discussed above with respect to claim 45.

#### 4. Claim 59

With respect to claim 59, claim 59 recites:

A program embodied on a computer-readable medium, comprising:

a code segment to create a license pack containing a set of one or more individual software licenses;

a code segment to assign a license pack ID to the license pack, the license pack ID uniquely identifying the license pack;

a code segment to associate the license pack ID with the particular license server; and

a code segment to digitally sign the license pack.

Appellant respectfully submits that Wyman in view of Christiano does not disclose or suggest a code segment to assign a license pack ID to the license pack, the license pack ID uniquely identifying the license pack, and a code segment to associate the license pack ID with the particular license server as recited in claim 59.

Appellant respectfully submits that, similar to the discussion above regarding claim 1, Christiano discusses a package license description that includes a package name field (see, col. 8, lines 57-61). The package name field stores the package name, which is the identifier of the package (see, col. 9, lines 1-2). However, the mere use of a package name does not disclose an ID uniquely identifying a license pack. A name is not necessarily unique. For example, multiple copies of the Microsoft® Word word processing application exist, each having the same files with the same file names. As such, these names do not uniquely identify the different files. There is nothing in Christiano that requires each package license description to have a unique name, nor is there any disclosure in Christiano of why each package license description would need a

unique name. Thus, Appellant respectfully submits that Christiano does not disclose an ID uniquely identifying a license pack as recited in claim 59.

With respect to Wyman, Wyman is not cited as curing, and does not cure, these deficiencies of Christiano.

For at least these reasons, Appellant respectfully submits that claim 59 is allowable over Wyman in view of Christiano.

## **5. Claim 68**

With respect to claim 68, claim 68 recites:

program embodied on a computer-readable medium, comprising:

- a code segment to receive a license pack from a license generator, the license pack containing a set of one or more individual software licenses;

- a code segment to validate the license pack;

- a code segment to store the software licenses;

- a code segment, responsive to a request for a software license from a client, to determine, based on a client executable image received from the client, whether the client is authentic and can receive a software license;

- a code segment to grant a software license to an authenticated client, the software license containing a license ID; and

- a code segment to associate the license ID with the authenticated client.

Appellant respectfully submits that Wyman in view of Christiano does not disclose or suggest a code segment, responsive to a request for a software license from a client, to determine, based on a client executable image received from the client, whether the client is authentic and can receive a software license as recited in claim 68.



Similar to the discussion above regarding claim 13, Appellant respectfully submits that there is no discussion or mention of a client executable image in Wyman or Christiano, much less any discussion or mention of a code segment, responsive to a request for a software license from a client, to determine, based on a client executable image received from the client, whether the client is authentic and can receive a software license as recited in claim 68. It appears from the June 16, 2005 Office Action at ¶4, p. 8, that Christiano is being relied on as teaching the client executable image of claim 68. However, as discussed above with respect to claim 83, Appellant respectfully submits that Christiano does not disclose or suggest a client image, and thus also does not disclose or suggest a client executable image as recited in claim 68.

Furthermore, claim 68 recites (emphasis added) to determine, based on a **client executable image** received from the client, whether the client is authentic and can receive a software license. In the June 16, 2005 Office Action at ¶ 8, p. 28, it was asserted that “A client is identified via a client image or identifier system (column 6, line 10 to column 7, line 30). In particular note column 6, line 64 to column 7, line 1.” Christiano at col. 6, line 64 to col. 7, line 1 recites “Thus, under a node-locked policy, a computer system 12 would be able to check out a license only if a license were available on the license server which matched that computer system’s node, address, serial number, user name, or other identifier”.

However, none of these items mentioned in the cited portion of Christiano is a client **executable** image. In Christiano, the “node-locked” policy allows only one computer system at a specific node on the license management system to use a designated software product (see, col. 6, lines 61-64). Thus, the computer

system's node is not a client **executable** image. A computer system's address is just an address, it is not a client **executable** image. A computer system's serial number is just a serial number, it is not a client **executable** image. A user name is just that – a user name; it is not a client **executable** image. The simple mentioning of “other identifier” does not teach a client **executable** image.

With respect to Wyman, Wyman is not cited as curing, and does not cure, these deficiencies of Christiano.

For at least these reasons, Appellant respectfully submits that claim 68 is allowable over Wyman in view of Christiano.

#### **6. Claim 69**

With respect to claim 69, claim 69 recites:

A client computer, comprising:  
a license cache to store one or more software licenses;  
a license requestor to request a software license from a license server;  
a challenge handler to handle an authenticity challenge from the license server, the challenge handler computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed;  
and  
whereupon authentication by the license server and granting of a software license, the license requestor receiving the software license from the license server and storing the software license in the license cache.

Appellant respectfully submits that Wyman in view of Christiano does not disclose or suggest a challenge handler to handle an authenticity challenge from the license server, the challenge handler computing a challenge response that

contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed as recited in claim 69.

In the June 16, 2005 Office Action at ¶ 8, p. 27, it was asserted that “A client is identified via a client image or identifier system (column 6, line 10 to column 7, line 30). In particular note column 6, line 64 to column 7, line 1.” Christiano at col. 6, line 64 to col. 7, line 1 recites “Thus, under a node-locked policy, a computer system 12 would be able to check out a license only if a license were available on the license server which matched that computer system’s node, address, serial number, user name, or other identifier”.

However, none of these items mentioned in the cited portion of Christiano is a client image. In Christiano, the “node-locked” policy allows only one computer system at a specific node on the license management system to use a designated software product (see, col. 6, lines 61-64). Thus, the computer system’s node is not a client image. A computer system’s address is just an address, it is not a client image. A computer system’s serial number is just a serial number, it is not a client image. A user name is just that – a user name; it is not a client image. The simple mentioning of “other identifier” does not teach a client image.

There is no discussion or mention in Christiano of a client image. In fact, a search through Christiano reveals that the word “image” is not present in Christiano, nor is “client image”. Accordingly, Appellant respectfully submits that the identifiers listed in the cited portion of Christiano do not disclose or suggest a client image, nor is a client image disclosed or suggested elsewhere in Christiano. As such, Appellant respectfully submits that Christiano cannot

disclose or suggest the client image as recited in claim 69, much less a challenge handler to handle an authenticity challenge from the license server, the challenge handler computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed as recited in claim 69.

With respect to Wyman, Wyman is not cited as curing, and does not cure, these deficiencies of Christiano.

For at least these reasons, Appellant respectfully submits that claim 69 is allowable over Wyman in view of Christiano.

#### **7. Claim 71**

With respect to claim 71, claim 71 recites:

A program embodied on a computer-readable medium, comprising:

a code segment to receive an authenticity challenge from a license server that distributes software licenses;

a code segment to compute a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed; and

a code segment to store the software license granted by the license server in an event that the client is deemed authentic.

Appellant respectfully submits that Wyman in view of Christiano does not disclose or suggest a code segment to compute a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed as recited in claim 71.

Similar to the discussion above regarding claim 69, none of the items mentioned in the cited portion of Christiano is a client image. In Christiano, the "node-locked" policy allows only one computer system at a specific node on the

license management system to use a designated software product (see, col. 6, lines 61-64). Thus, the computer system's node is not a client image. A computer system's address is just an address, it is not a client image. A computer system's serial number is just a serial number, it is not a client image. A user name is just that — a user name; it is not a client image. The simple mentioning of "other identifier" does not teach a client image.

There is no discussion or mention in Christiano of a client image. In fact, a search through Christiano reveals that the word "image" is not present in Christiano, nor is "client image". Accordingly, Appellant respectfully submits that the identifiers listed in the cited portion of Christiano do not disclose or suggest a client image, nor is a client image disclosed or suggested elsewhere in Christiano. As such, Appellant respectfully submits that Christiano cannot disclose or suggest the client image as recited in claim 71, much less a code segment to compute a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed as recited in claim 71.

With respect to Wyman, Wyman is not cited as curing, and does not cure, these deficiencies of Christiano.

For at least these reasons, Appellant respectfully submits that claim 71 is allowable over Wyman in view of Christiano.

## **8. Claims 72-75**

With respect to claim 72, claim 72 recites:

A data structure embodied on a computer-readable media,  
comprising:

a license pack table to record information pertaining to one or more license packs, the license pack table being indexed by license pack IDs that uniquely identify corresponding individual license packs, each license pack containing one or more software licenses;

a client assignment table to record information pertaining to software licenses that are assigned to clients, the client assignment table being indexed by license IDs that identify individual software licenses, the client assignment table further having the license pack IDs of the license packs from which the corresponding software licenses are issued; and

the license pack table and the client assignment table being correlated via the license pack IDs contained in each table.

Appellant respectfully submits that Wyman in view of Christiano does not disclose or suggest a license pack table to record information pertaining to one or more license packs, the license pack table being indexed by license pack IDs that uniquely identify corresponding individual license packs, each license pack containing one or more software licenses as recited in claim 72.

Appellant respectfully submits that, similar to the discussion above regarding claim 1, Christiano discusses a package license description that includes a package name field (see, col. 8, lines 57-61). The package name field stores the package name, which is the identifier of the package (see, col. 9, lines 1-2). However, the mere use of a package name does not disclose an ID uniquely identifying a license pack. A name is not necessarily unique. For example, multiple copies of the Microsoft® Word word processing application exist, each having the same files with the same file names. As such, these names do not uniquely identify the different files. There is nothing in Christiano that requires each package license description to have a unique name, nor is there any disclosure in Christiano of why each package license description would need a unique name. Thus, Appellant respectfully submits that Christiano does not disclose an ID uniquely identifying a license pack as recited in claim 72.

With respect to Wyman, Wyman is not cited as curing, and does not cure, these deficiencies of Christiano.

For at least these reasons, Appellant respectfully submits that claim 72 is allowable over Wyman in view of Christiano.

Given that claims 73-75 depend from claim 72, Appellant respectfully submits that claims 73-75 are likewise allowable over Wyman in view of Christiano for at least the reasons discussed above with respect to claim 72.

**C. Rejection under 35 U.S.C. §103(a) over U.S. Patent No. 5,745,879 to Wyman in view of U.S. Patent No. 5,671,412 to Christiano and in further view of U.S. Patent No. 5,138,712 to Corbin.**

Claims 15-21, 23-24, 26-27, 29, 55-57, 60-61, and 63-65 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,745,879 to Wyman (hereinafter “Wyman”) in view of U.S. Patent No. 5,671,412 to Christiano (hereinafter “Christiano”) and in further view of U.S. Patent No. 5,138,712 to Corbin (hereinafter “Corbin”).

**1. Claims 15-20**

With respect to claim 15, claim 15 recites:

A computer-implemented method comprising:  
receiving a request for a software license from a particular license server;  
creating a license pack containing a set of one or more individual software licenses;  
assigning a license pack ID to the license pack, the license pack ID uniquely identifying the license pack;  
associating the license pack ID with the particular license server;

digitally signing the license pack; and  
issuing the signed license pack to the particular license server.

Appellant respectfully submits that Wyman in view of Christiano and Corbin does not disclose or suggest assigning a license pack ID to the license pack, the license pack ID uniquely identifying the license pack, and associating the license pack ID with the particular license server as recited in claim 15.

Appellant respectfully submits that, similar to the discussion above regarding claim 1, Christiano discusses a package license description that includes a package name field (see, col. 8, lines 57-61). The package name field stores the package name, which is the identifier of the package (see, col. 9, lines 1-2). However, the mere use of a package name does not disclose an ID uniquely identifying a license pack. A name is not necessarily unique. For example, multiple copies of the Microsoft® Word word processing application exist, each having the same files with the same file names. As such, these names do not uniquely identify the different files. There is nothing in Christiano that requires each package license description to have a unique name, nor is there any disclosure in Christiano of why each package license description would need a unique name. Thus, Appellant respectfully submits that Christiano does not disclose an ID uniquely identifying a license pack as recited in claim 15.

With respect to Wyman and Corbin, Wyman and Corbin are not cited as curing, and do not cure, these deficiencies of Christiano.

For at least these reasons, Appellant respectfully submits that claim 15 is allowable over Wyman in view of Christiano and Corbin.

Given that claims 16-20 depend from claim 15, Appellant respectfully submits that claims 16-20 are likewise allowable over Wyman in view of



Christiano and Corbin for at least the reasons discussed above with respect to claim 15.

## **2. Claims 21, 23-24, 26-27, and 29**

With respect to claim 21, claim 21 recites:

A computer-implemented method comprising:  
receiving a request for a software license from a particular client;  
determining an authenticity of the particular client, wherein the determining includes:  
maintaining a set of client images;  
receiving a client software ID from the particular client; and  
comparing the client software ID to the client images to evaluate whether the client is authentic;  
selecting a software license from a pack of software licenses that is appropriate for the particular client, the software license having an associated license ID;  
associating the license ID with the particular client; and  
granting the software license to the particular client.

Appellant respectfully submits that Wyman in view of Christiano and Corbin does not disclose or suggest maintaining a set of client images, receiving a client software ID from the particular client, and comparing the client software ID to the client images to evaluate whether the client is authentic as recited in claim 21.

In the June 16, 2005 Office Action at ¶ 8, p. 27, it was asserted that “A client is identified via a client image or identifier system (column 6, line 10 to column 7, line 30). In particular note column 6, line 64 to column 7, line 1.” Christiano at col. 6, line 64 to col. 7, line 1 recites “Thus, under a node-locked policy, a computer system 12 would be able to check out a license only if a license

were available on the license server which matched that computer system's node, address, serial number, user name, or other identifier".

However, none of these items mentioned in the cited portion of Christiano is a client image. In Christiano, the "node-locked" policy allows only one computer system at a specific node on the license management system to use a designated software product (see, col. 6, lines 61-64). Thus, the computer system's node is not a client image. A computer system's address is just an address, it is not a client image. A computer system's serial number is just a serial number, it is not a client image. A user name is just that -- a user name; it is not a client image. The simple mentioning of "other identifier" does not teach a client image.

There is no discussion or mention in Christiano of client images. In fact, a search through Christiano reveals that the word "image" is not present in Christiano, nor is "client image". Accordingly, Appellant respectfully submits that the identifiers listed in the cited portion of Christiano do not disclose or suggest client images, nor are client images disclosed or suggested elsewhere in Christiano. As such, Appellant respectfully submits that Christiano cannot disclose or suggest the client images as recited in claim 21, much less maintaining a set of client images, receiving a client software ID from the particular client, and comparing the client software ID to the client images to evaluate whether the client is authentic as recited in claim 21.

Corbin and Wyman are not cited as curing, and do not cure, the deficiencies of Christiano discussed above with respect to claim 21. For at least these reasons,

Appellant respectfully submits that claim 21 is allowable over Wyman in view of Christiano and Corbin.

Given that claims 23, 24, 26, 27, and 29 depend from claim 21, Appellant respectfully submits that claims 23, 24, 26, 27, and 29 are likewise allowable over Wyman in view of Christiano and Corbin for at least the reasons discussed above with respect to claim 21.

### **3. Claims 55-57**

With respect to claim 55, claim 55 recites:

A license generator for issuing packs of software licenses to authorized license servers, comprising:

a request handler to receive a request from a license server for a license pack;

a license producer responsive to the request received by the request handler to generate a license pack containing a set of one or more individual software licenses; and

the license producer assigning a license pack ID to the license pack, associating the license pack ID with the license server, and digitally signing the license pack, the license pack ID uniquely identifying the license pack.

Appellant respectfully submits that Wyman in view of Christiano and Corbin does not disclose or suggest the license producer assigning a license pack ID to the license pack, associating the license pack ID with the license server, and digitally signing the license pack, the license pack ID uniquely identifying the license pack as recited in claim 55.

Appellant respectfully submits that, similar to the discussion above regarding claim 1, Christiano discusses a package license description that includes a package name field (see, col. 8, lines 57-61). The package name field stores the package name, which is the identifier of the package (see, col. 9, lines 1-2).

However, the mere use of a package name does not disclose an ID uniquely identifying a license pack. A name is not necessarily unique. For example, multiple copies of the Microsoft® Word word processing application exist, each having the same files with the same file names. As such, these names do not uniquely identify the different files. There is nothing in Christiano that requires each package license description to have a unique name, nor is there any disclosure in Christiano of why each package license description would need a unique name. Thus, Appellant respectfully submits that Christiano does not disclose an ID uniquely identifying a license pack as recited in claim 55.

Corbin and Wyman are not cited as curing, and do not cure, these deficiencies of Christiano. For at least these reasons, Appellant respectfully submits that claim 55 is allowable over Wyman in view of Christiano and Corbin.

Given that claims 56 and 57 depend from claim 55, Appellant respectfully submits that claims 56 and 57 are likewise allowable over Wyman in view of Christiano and Corbin for at least the reasons discussed above with respect to claim 55.

#### **4. Claims 60-61 and 63-65**

With respect to claim 60, claim 60 recites:

A license server for issuing individual software licenses from a software pack received from a licensing clearinghouse, comprising:

- a license store to store the software pack of individual software licenses, each software license having an associated license ID;

- a request handler to receive a request for a software license from a client;

a client authenticating module to determine, based on a client image received from the client, whether the client is authentic and can receive a software license; and

a granting module to grant a software license from the license store to an authenticated client and to associate the license ID with the authenticated client.

Appellant respectfully submits that Wyman in view of Christiano and Corbin does not disclose or suggest a client authenticating module to determine, based on a client image received from the client, whether the client is authentic and can receive a software license as recited in claim 60.

In the June 16, 2005 Office Action at ¶ 8, p. 27, it was asserted that “A client is identified via a client image or identifier system (column 6, line 10 to column 7, line 30). In particular note column 6, line 64 to column 7, line 1.” Christiano at col. 6, line 64 to col. 7, line 1 recites “Thus, under a node-locked policy, a computer system 12 would be able to check out a license only if a license were available on the license server which matched that computer system’s node, address, serial number, user name, or other identifier”.

However, none of these items mentioned in the cited portion of Christiano is a client image. In Christiano, the “node-locked” policy allows only one computer system at a specific node on the license management system to use a designated software product (see, col. 6, lines 61-64). Thus, the computer system’s node is not a client image. A computer system’s address is just an address, it is not a client image. A computer system’s serial number is just a serial number, it is not a client image. A user name is just that – a user name; it is not a client image. The simple mentioning of “other identifier” does not teach a client image.

There is no discussion or mention in Christiano of a client image. In fact, a search through Christiano reveals that the word “image” is not present in Christiano, nor is “client image”. Accordingly, Appellant respectfully submits that the identifiers listed in the cited portion of Christiano do not disclose or suggest a client image, nor is a client image disclosed or suggested elsewhere in Christiano. As such, Appellant respectfully submits that Christiano cannot disclose or suggest the client image as recited in claim 60, much less a client authenticating module to determine, based on a client image received from the client, whether the client is authentic and can receive a software license as recited in claim 60.

Corbin and Wyman are not cited as curing, and do not cure, these deficiencies of Christiano. For at least these reasons, Appellant respectfully submits that claim 60 is allowable over Wyman in view of Christiano and Corbin.

Given that claims 61 and 63-65 depend from claim 60, Appellant respectfully submits that claims 61 and 63-65 are likewise allowable over Wyman in view of Christiano and Corbin for at least the reasons discussed above with respect to claim 60.

**D. Rejection under 35 U.S.C. §103(a) over U.S. Patent No. 5,671,412 to Christiano in view of U.S. Patent No. 6,049,612 to Fielder et al.**

Claims 79-82 and 86-88 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,671,412 to Christiano (hereinafter “Christiano”) in view of U.S. Patent No. 6,049,612 to Fielder et al. (hereinafter “Fielder”).

**1. Claims 79-82**

With respect to claim 79, claim 79 recites:

A client computer, comprising:  
memory for storing information corresponding to one or more received software licenses;  
a license requestor to request a software license from a license server;  
a challenge handler to handle an authenticity challenge from the license server, the challenge handler computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed, wherein the challenge contains a random number, and the challenge handler computes the challenge response by concatenating the random number with the client image to form a concatenated value and hashing the concatenated value; and  
wherein, upon authentication by the license server and granting of a software license, the license requestor receiving the software license from the license server saves in the memory information corresponding to the received license.

Appellant respectfully submits that Christiano in view of Fielder does not disclose or suggest a challenge handler to handle an authenticity challenge from the license server, the challenge handler computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed, wherein the challenge contains a random number, and the challenge handler computes the challenge response by concatenating the random number with the client image to form a concatenated value and hashing the concatenated value as recited in claim 79.

In the June 16, 2005 Office Action at ¶ 8, p. 27, it was asserted that “A client is identified via a client image or identifier system (column 6, line 10 to column 7, line 30). In particular note column 6, line 64 to column 7, line 1.”

Christiano at col. 6, line 64 to col. 7, line 1 recites “Thus, under a node-locked policy, a computer system 12 would be able to check out a license only if a license were available on the license server which matched that computer system’s node, address, serial number, user name, or other identifier”.

However, none of these items mentioned in the cited portion of Christiano is a client image. In Christiano, the “node-locked” policy allows only one computer system at a specific node on the license management system to use a designated software product (see, col. 6, lines 61-64). Thus, the computer system’s node is not a client image. A computer system’s address is just an address, it is not a client image. A computer system’s serial number is just a serial number, it is not a client image. A user name is just that – a user name; it is not a client image. The simple mentioning of “other identifier” does not teach a client image.

There is no discussion or mention in Christiano of a client image. In fact, a search through Christiano reveals that the word “image” is not present in Christiano, nor is “client image”. Accordingly, Appellant respectfully submits that the identifiers listed in the cited portion of Christiano do not disclose or suggest a client image, nor is a client image disclosed or suggested elsewhere in Christiano. As such, Appellant respectfully submits that Christiano cannot disclose or suggest the client image as recited in claim 79, much less a challenge handler computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed as recited in claim 79.



Fielder is not cited as curing, and does not cure, these deficiencies of Christiano. For at least these reasons, Appellant respectfully submits that claim 79 is allowable over Christiano in view of Fielder.

Given that claims 80-82 depend from claim 79, Appellant respectfully submits that claims 80-82 are likewise allowable over Christiano in view of Fielder for at least the reasons discussed above with respect to claim 79.

## **2. Claims 86-88**

With respect to claim 86, claim 86 recites:

A client computer, comprising:  
means for storing information corresponding to one or more received software licenses;  
means for requesting a software license from a license server;  
means for handling an authenticity challenge from the license server, the means for handling computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed, wherein the challenge contains a random number, and the means for handling computes the challenge response by concatenating the random number with the client image to form a concatenated value and hashing the concatenated value; and  
wherein, upon authentication by the license server and granting of a software license, the means for requesting receiving the software license from the license server saves in the memory information corresponding to the received license.

Appellant respectfully submits that Christiano in view of Fielder does not disclose or suggest means for handling an authenticity challenge from the license server, the means for handling computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed, wherein the challenge contains a random number, and the means for handling computes the challenge response by concatenating the

random number with the client image to form a concatenated value and hashing the concatenated value as recited in claim 86.

In the June 16, 2005 Office Action at ¶ 8, p. 27, it was asserted that “A client is identified via a client image or identifier system (column 6, line 10 to column 7, line 30). In particular note column 6, line 64 to column 7, line 1.” Christiano at col. 6, line 64 to col. 7, line 1 recites “Thus, under a node-locked policy, a computer system 12 would be able to check out a license only if a license were available on the license server which matched that computer system’s node, address, serial number, user name, or other identifier”.

However, none of these items mentioned in the cited portion of Christiano is a client image. In Christiano, the “node-locked” policy allows only one computer system at a specific node on the license management system to use a designated software product (see, col. 6, lines 61-64). Thus, the computer system’s node is not a client image. A computer system’s address is just an address, it is not a client image. A computer system’s serial number is just a serial number, it is not a client image. A user name is just that – a user name; it is not a client image. The simple mentioning of “other identifier” does not teach a client image.

There is no discussion or mention in Christiano of a client image. In fact, a search through Christiano reveals that the word “image” is not present in Christiano, nor is “client image”. Accordingly, Appellant respectfully submits that the identifiers listed in the cited portion of Christiano do not disclose or suggest a client image, nor is a client image disclosed or suggested elsewhere in Christiano. As such, Appellant respectfully submits that Christiano cannot

disclose or suggest the client image as recited in claim 86, much less a means for handling computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed as recited in claim 86.

Fielder is not cited as curing, and does not cure, these deficiencies of Christiano. For at least these reasons, Appellant respectfully submits that claim 86 is allowable over Christiano in view of Fielder.

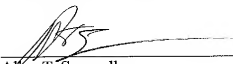
Given that claims 87 and 88 depend from claim 86, Appellant respectfully submits that claims 87 and 88 are likewise allowable over Christiano in view of Fielder for at least the reasons discussed above with respect to claim 86.

### **Conclusion**

The Office's basis and supporting rationale for the § 102(a) and § 103(a) rejections is not supported by the teaching of the cited references. Appellant respectfully requests that the rejections be overturned and that pending claims 1-6, 10, 13-21, 23, 24, 26, 27, 29, 45, 47-57, 59-61, 63-65, 68, 69, 71-75, and 79-88 be allowed to issue.

Respectfully Submitted,

Dated: 5/15/06

By:   
Alan T. Sponseller  
Lee & Hayes, PLLC  
Reg. No. 38,318  
(509) 324-9256 ext. 215

**(8) Appendix of Appealed Claims**

1. A computer-implemented method comprising:  
creating a license pack at a license generator, the license pack containing a set of one or more individual software licenses;  
signing the license pack with a digital signature of the license generator;  
associating an ID of the license pack with a license server, the ID uniquely identifying the license pack;  
maintaining the association at the license generator;  
issuing the license pack to the license server;  
verifying, at the license server, the license generator's digital signature on the license pack; and  
distributing the software licenses contained in the license pack from the license server to corresponding clients.

2. A computer-implemented method as recited in claim 1, further comprising creating a license pack containing a predefined number of software licenses.

3. A computer-implemented method as recited in claim 1, further comprising:  
creating a license pack ID at the license generator; and  
evaluating the license pack ID at the license server.

4. A computer-implemented method as recited in claim 1, further comprising:

encrypting the license pack at the license generator; and

decrypting the license pack at the license server.

5. A computer-implemented method as recited in claim 1, further comprising creating a license pack that is tailored to a particular operating platform of the clients.

6. A computer-implemented method as recited in claim 1, further comprising determining an authenticity of an individual client prior to distributing the software license to that individual client.

10. A computer-implemented method as recited in claim 1, wherein the license pack has a license pack ID, and further comprising granting additional licenses for the license pack having the same license pack ID.

13. A computer-implemented method for distributing software licenses to clients so that the clients may legally execute underlying software to which the software licenses pertain, the computer-implemented method comprising electronically issuing the software licenses as digital certificates that are distributed in one-to-one correlation with individual clients and traced to an issuing authority, and issuing a software license to a particular client only if a

client executable image received from the client matches a stored client executable image for the particular client.

14. A computer-readable medium having computer readable instructions for performing the method as recited in claim 13.

15. A computer-implemented method comprising:  
receiving a request for a software license from a particular license server;  
creating a license pack containing a set of one or more individual software licenses;

assigning a license pack ID to the license pack, the license pack ID uniquely identifying the license pack;

associating the license pack ID with the particular license server;

digitally signing the license pack; and

issuing the signed license pack to the particular license server.

16. A computer-implemented method as recited in claim 15, further comprising creating a license pack containing a predefined number of software licenses.

17. A computer-implemented method as recited in claim 15, further comprising creating a license pack that includes a platform type indicating a type of operating platform for which the software licenses can be used.

18. A computer-implemented method as recited in claim 15, further comprising creating a license pack that includes a predefined number of software licenses, a platform type indicating a type of operating platform for which the software licenses can be used, an expiration date indicating a date on which the software licenses will expire, and a product ID that identifies a product with which the software licenses can be used.

19. A computer-implemented method as recited in claim 15, further comprising encrypting the license pack.

20. A computer-readable medium having computer readable instructions for performing the method as recited in claim 15.

21. A computer-implemented method comprising:  
receiving a request for a software license from a particular client;  
determining an authenticity of the particular client, wherein the determining includes:

maintaining a set of client images;

receiving a client software ID from the particular client; and

comparing the client software ID to the client images to evaluate whether the client is authentic;

selecting a software license from a pack of software licenses that is appropriate for the particular client, the software license having an associated license ID;

associating the license ID with the particular client; and  
granting the software license to the particular client.

23. A computer-implemented method as recited in claim 21, further comprising granting the software license as containing the license ID, a platform type indicating a type of platform in use by the particular client, an issue date indicates a date on which the license is issued to the client, an expiration date that indicates a date on which the software license will expire, a product ID that identifies a product with which the software licenses can be used, a client ID that identifies the particular client, and a version of the software license.

24. A computer-implemented method as recited in claim 21, wherein determining the authenticity comprises:

receiving a client software ID from the particular client; and

evaluating the client software ID to determine whether the client is authentic.

26. A computer-implemented method as recited in claim 21, further comprising:

determining a platform of the particular client; and

selecting the software license as is appropriate for the platform of the particular client.



27. A computer-implemented method as recited in claim 21, further comprising encrypting the software license using a public key of the particular client.

29. A computer-readable medium having computer readable instructions for performing the method as recited in claim 21.

45. A system for licensing software, comprising:

a license generator to create a license pack containing a set of one or more individual software licenses, the license generator digitally signing the license pack with a digital signature;

a license server remote from, but operatively coupled to, the license generator to receive the license pack from the license generator, the license server verifying the license generator's digital signature on the license pack and storing the individual licenses for subsequent distribution to individual clients; and

wherein the license generator is further to assign a license pack ID to the license pack and keep an association of the license pack ID with the license server, the license pack ID uniquely identifying the license pack.

47. A system as recited in claim 45, wherein the license generator encrypts the license pack using a public key of the license server.

48. A system as recited in claim 45, wherein the license pack contains a preset number of software licenses.

49. A system as recited in claim 45, wherein the license pack identifies a type of operating platform for which the software licenses can be used.

50. A system as recited in claim 45, wherein the license pack comprises at least one of the following items:

a predefined number of software licenses;

a platform type indicating a type of operating platform for which the software licenses can be used;

an expiration date that indicates a date on which the software licenses will expire; and

a product ID that identifies a product with which the software licenses can be used.

51. A system as recited in claim 45, wherein the license server selects a software license from the license pack and grants the software license to a client, the software license having a license ID and the license server associating the license ID with the client.

52. A system as recited in claim 45, wherein the license server challenges an authenticity of a client prior to granting a software license from the license pack to the client.

53. A system as recited in claim 45, wherein the license server grants a software license server to a particular client, the license server encrypting the software license using a public key of the particular client.

54. A system as recited in claim 45, wherein the license server distributes the software licenses to the individual clients via one or more intermediate servers.

55. A license generator for issuing packs of software licenses to authorized license servers, comprising:

- a request handler to receive a request from a license server for a license pack;

- a license producer responsive to the request received by the request handler to generate a license pack containing a set of one or more individual software licenses; and

- the license producer assigning a license pack ID to the license pack, associating the license pack ID with the license server, and digitally signing the license pack, the license pack ID uniquely identifying the license pack.

56. A license generator as recited in claim 55, wherein the license pack contains a predefined number of software licenses, a platform type indicating a type of operating platform for which the software licenses can be used, an expiration date that indicates a date on which the software licenses will expire, and

a product ID that identifies a product with which the software licenses can be used.

57. A license generator as recited in claim 55, wherein the license producer encrypts the license pack using a public key of the license server.

59. A program embodied on a computer-readable medium, comprising:

- a code segment to create a license pack containing a set of one or more individual software licenses;
- a code segment to assign a license pack ID to the license pack, the license pack ID uniquely identifying the license pack;
- a code segment to associate the license pack ID with the particular license server; and
- a code segment to digitally sign the license pack.

60. A license server for issuing individual software licenses from a software pack received from a licensing clearinghouse, comprising:

- a license store to store the software pack of individual software licenses, each software license having an associated license ID;
- a request handler to receive a request for a software license from a client;
- a client authenticating module to determine, based on a client image received from the client, whether the client is authentic and can receive a software license; and

a granting module to grant a software license from the license store to an authenticated client and to associate the license ID with the authenticated client.

61. A license server as recited in claim 60, wherein the authenticating module determines an operating platform of the client.

63. A license server as recited in claim 60, wherein the granting module encrypts the software license using a public key of the authenticated client.

64. A license server as recited in claim 60, wherein the software license contains at least one of the following items:

- a version indicator of a software license;

- a license ID;

- a client ID that identifies the authenticated client;

- an issue date on which the license is issued to the client;

- a platform type of the client's operating platform for which the software license can be used;

- an expiration date on which the software license will expire; and

- a product ID that identifies a product with which the software licenses can be used.

65. A license server as recited in claim 60, further comprising a license pack table to store information pertaining to the license pack that is stored in the license store.

68. A program embodied on a computer-readable medium, comprising:

- a code segment to receive a license pack from a license generator, the license pack containing a set of one or more individual software licenses;
- a code segment to validate the license pack;
- a code segment to store the software licenses;
- a code segment, responsive to a request for a software license from a client, to determine, based on a client executable image received from the client, whether the client is authentic and can receive a software license;
- a code segment to grant a software license to an authenticated client, the software license containing a license ID; and
- a code segment to associate the license ID with the authenticated client.

69. A client computer, comprising:

- a license cache to store one or more software licenses;
- a license requestor to request a software license from a license server;
- a challenge handler to handle an authenticity challenge from the license server, the challenge handler computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed; and

whereupon authentication by the license server and granting of a software license, the license requestor receiving the software license from the license server and storing the software license in the license cache.

71. A program embodied on a computer-readable medium, comprising:

- a code segment to receive an authenticity challenge from a license server that distributes software licenses;
- a code segment to compute a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed; and
- a code segment to store the software license granted by the license server in an event that the client is deemed authentic.

72. A data structure embodied on a computer-readable media, comprising:

- a license pack table to record information pertaining to one or more license packs, the license pack table being indexed by license pack IDs that uniquely identify corresponding individual license packs, each license pack containing one or more software licenses;

- a client assignment table to record information pertaining to software licenses that are assigned to clients, the client assignment table being indexed by license IDs that identify individual software licenses, the client assignment table further having the license pack IDs of the license packs from which the corresponding software licenses are issued; and

- the license pack table and the client assignment table being correlated via the license pack IDs contained in each table.

73. A data structure as recited in claim 72, wherein the license pack table contains the following table fields:

- a license pack ID field to hold the license pack ID;

- a quantity field to hold a number representative of how many software licenses are contained in the license pack;

- a platform type field to hold a type of operating platform for which the software licenses in the license pack can be used;

- an expiration date field to hold a date on which the software licenses in the license pack will expire; and

- a product ID field to hold a product ID that identifies a product with which the software licenses in the license pack can be used.

74. A data structure as recited in claim 73, wherein the license pack table also contains a number assigned field to hold a number representative of how many of the software licenses have been assigned to clients.

75. A data structure as recited in claim 72, wherein the client assignment table contains the following table fields:

- a license ID field to hold the license ID;

- a license pack ID field to hold the license pack ID;

- a client ID to hold an identifier of a client to which the software license is granted; and

- an issue date to hold a date on which the software license is issued to the client.



79. A client computer, comprising:

memory for storing information corresponding to one or more received software licenses;

a license requestor to request a software license from a license server;

a challenge handler to handle an authenticity challenge from the license server, the challenge handler computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed, wherein the challenge contains a random number, and the challenge handler computes the challenge response by concatenating the random number with the client image to form a concatenated value and hashing the concatenated value; and

wherein, upon authentication by the license server and granting of a software license, the license requestor receiving the software license from the license server saves in the memory information corresponding to the received license.

80. The client computer of claim 79, wherein the memory comprises non-volatile memory.

81. The client computer of claim 79, wherein the saved information corresponding to the received license comprises the received license.

82. The client computer of claim 79, wherein the saved information corresponding to the received license comprises an indication that a specific license was granted.

83. A computer-readable medium, having computer readable instructions for:

requesting a software license from a license server;

receiving an authenticity challenge from the license server;

computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed; and

receiving, upon authentication by the license server and granting of a software license, the software license from the license server and storing information corresponding to the software license in a memory.

84. The computer-readable medium of claim 83, wherein the stored information corresponding to the received software license comprises the received software license.

85. The computer-readable medium of claim 83, wherein the stored information corresponding to the received software license comprises an indication that a specific license was granted.

86. A client computer, comprising:

means for storing information corresponding to one or more received software licenses;

means for requesting a software license from a license server;

means for handling an authenticity challenge from the license server, the means for handling computing a challenge response that contains a client image that can be used by the license server to evaluate whether the client is authentic and can be licensed, wherein the challenge contains a random number, and the means for handling computes the challenge response by concatenating the random number with the client image to form a concatenated value and hashing the concatenated value; and

wherein, upon authentication by the license server and granting of a software license, the means for requesting receiving the software license from the license server saves in the memory information corresponding to the received license.

87. The client computer of claim 86, wherein the saved information corresponding to the received license comprises the received license.

88. The client computer of claim 86, wherein the saved information corresponding to the received license comprises an indication that a specific license was granted.

**(9) Appendix of Evidence Submitted**

None.

**(10) Appendix of Related Proceedings**

None.